UDACITY

Logout

PROJECT

PID Controller

A part of the Self Driving Car Engineer Nanodegree Program

PROJECT REVIEW | CODE REVIEW | NOTES

Meets Specifications

SHARE YOUR ACCOMPLISHMENT

Awesome job implementing your PID controller. I look forward to how you approach your MPC controller to help smooth out the control scheme!

Keep up the hard work and stay Udacious!

Compilation

✓ **Code must compile without errors with `cmake` and `make`.**

Given that we've made CMakeLists.txt as general as possible, it's recommend that you do not change it unless you can guarantee that your changes will still compile on any platform.

Your code complied without any errors!

Implementation

✓ **It's encouraged to be creative, particularly around hyperparameter tuning/optimization. However, the base algorithm should follow what's presented in the lessons.**

Great job with your implementation of the controller. Code is very lean!

Reflection

✓ **Student describes the effect of the P, I, D component of the PID algorithm in their implementation. Is it what you expected?**

Visual aids are encouraged, i.e. record of a small video of the car in the simulator and describe what each component is set to.

All aspects of the PID controller are discussed and explained. Excellent!

✓ **Student discusses how they chose the final hyperparameters (P, I, D coefficients). This could be have been done through manual tuning, twiddle, SGD, or something else, or a combination!**

Great job tuning your parameters!

Here is another resource to learn more about other tuning methods!

PID Tuning Resource

Simulation

✓ **No tire may leave the drivable portion of the track surface. The car may not pop up onto ledges or roll over any surfaces that would otherwise be considered unsafe (if humans were in the vehicle).**

The car made it around the track without leaving the road!

DOWNLOAD PROJECT

RETURN TO PATH